

# 10 - Linguagem DTD (**D**ocument **T**ype **D**efinition)

Baseado nos slides do professor Paulo Trigo  
Todas as alterações são da responsabilidade do professor António Teófilo  
e do professor Diogo Remédios

# O que é o DTD?

- Antes de responder é preciso estabelecer a diferença entre
  - ❑ documento XML: bem formado e válido
- Um documento XML diz-se bem formado
  - ❑ se respeitar as regras da linguagem XML
  - ❑ e.g. “o valor dos atributos é delimitado por aspas ou plicas”,
- Um documento XML diz-se válido
  - ❑ se a sua escrita respeitar uma determinada gramática
  - ❑ e.g. “o elemento `listaDeBares` tem zero ou mais elementos `bar`”
  - ❑ ... a marca `bar` só se pode escrever descendendo de `listaDeBares`!
- O DTD é uma linguagem para especificar gramáticas
  - ❑ O que a escrita de um documento XML deve respeitar
  - ❑ ... o documento diz-se válido se respeita a gramática para ele definida!
  - ❑ ... no entanto, o DTD tem uma sintaxe própria (diferente do XML)

e.g. ?????? O que é isto?

“e.g.” é uma abreviatura da frase em Latin: *exempli gratia*, que significa “por exemplo”

“i.e.” é uma abreviatura da frase em Latin: *id est*, que significa “isto é”

# Porquê usar o DTD?

- O DTD favorece a troca de documentos XML
  - ❑ documentos com mesmo DTD são tratados pelas mesmas aplicações
- O DTD pode especificar:
  - ❑ os elementos, atributos e entidades,
  - ❑ o contexto em que cada uma daquelas componentes se pode utilizar,
  - ❑ os atributos com valor único e as referências para esses atributos,
  - ❑ definições condicionais
- Um analisador (“parser”) confronta o documento XML com seu DTD
  - ❑ e pode rejeitar um documento que não esteja de acordo com o DTD
  - ❑ ... validar assume que “o que não está no DTD não se pode usar”
- Com o DTD cada documento XML
  - ❑ Pode transportar a descrição do seu próprio formato!

# ... uma lista de bares em XML

Na lista de bares  
constam  
diversos bares!

Cada bar tem um  
nome e serve  
pelo menos um  
tipo de cerveja!

Cada cerveja  
tem um nome e  
um preço.

```
<?xml version = "1.0" encoding="ISO-8859-1" standalone="yes"?>

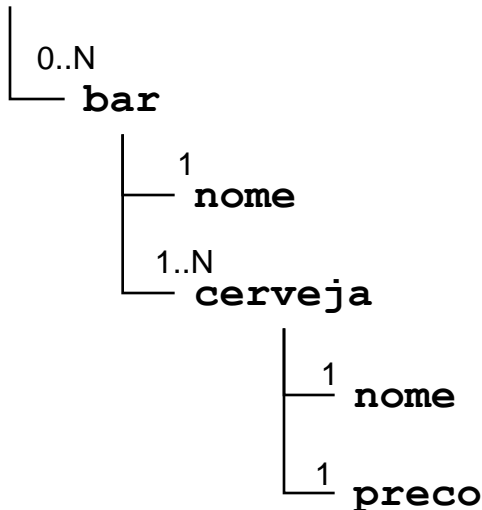
<listaDeBares>
  <bar>
    <nome>O Bar do Inglês</nome>
    <cerveja>
      <nome>Bud</nome>
      <preco>2.50</preco>
    </cerveja>
    <cerveja>
      <nome>Miller</nome>
      <preco>3.00</preco>
    </cerveja>
  </bar>
  <bar>
    <nome>O Pinguim</nome>
    <cerveja>
      <nome>Antarctica</nome>
      <preco>2.00</preco>
    </cerveja>
  </bar>
</listaDeBares>
```

# A estrutura daquela “lista de bares”

três formas de a descrever

V1

listaDeBares



V2

```
listaDeBares = bar*
bar = nome cerveja+
nome = string
cerveja = nome preco
preco = string
```

Tipo DTD

V3

```
<listaDeBares>
  <bar> *
    <nome>text</nome>
    <cerveja> +
      <nome>text</nome>
      <preco>text</preco>
    </cerveja>
  </bar>
</listaDeBares>
```

As tags de inicio podem ser complementadas com um indicador de cardinalidade..

Convenção:

? ≡ zero (0) ou uma (1) ocorrência  
\* ≡ zero (0) ou mais ocorrências  
+ ≡ uma (1) ou mais ocorrências  
| = “ou” alternativa entre uma coisa ou

outra

string: sequência de caracteres

# Como partilhar a minha “lista de bares”!


- Garantindo que o documento tem um formato que todos entendem
  - ❑ ... mas o XML não impõe um formato
  - ❑ a mesma informação pode ser representada com diferentes formatos!

```
<?xml version = "1.0" encoding="ISO-8859-1" standalone="yes"?>

<listaDeBares>
  <bar><nome>O Bar Inglês</nome></bar>
  <bar><nome>O Pinguim</nome></bar>

  <cerveja nome="Bud"><preco>2.50</preco></cerveja>
  <cerveja nome="Miller"><preco>3.00</preco></cerveja>
  <cerveja nome="Antarctica"><preco>2.00</preco></cerveja>

  <bar_cerveja oBar="O Bar Inglês" aCerveja="Bud"/>
  <bar_cerveja oBar="O Bar Inglês" aCerveja="Miller"/>
  <bar_cerveja oBar="O Pinguim" aCerveja="Antarctica"/>
</listaDeBares>
```



Uma lista de bares  
com uma estrutura  
completamente  
diferente!

Embora com a  
mesma  
informação...

# ... uma especificação DTD da “lista de bares” original

da especificação anterior para a especificação DTD

**listaDeBares** = bar\*

**bar** = nome *cerveja*+

**nome** = *string*

**cerveja** = nome **preco**

**preco** = *string*

<!ELEMENT **listaDeBares** (bar\*) >

<!ELEMENT **bar** (nome, *cerveja*+) >

<!ELEMENT **nome** (#PCDATA) >

<!ELEMENT **cerveja** (nome, **preco**) >

<!ELEMENT **preco** (#PCDATA) >

Convenção:

? ≡ zero (0) ou uma (1) ocorrência

\* ≡ zero (0) ou mais ocorrências

+ ≡ uma (1) ou mais ocorrências

especificação DTD  
da estrutura de um  
elemento XML

elemento  
apenas contém  
sequência de  
caracteres

A , ((B,C)\*, D+, (F, (G)+, H)?)

# Integrar a especificação DTD com o documento XML

- ... a linguagem DTD (“Document Type Definition”)
  - permite especificar uma estrutura XML
- O DTD pode estar incluído (“inline”) no próprio documento XML
  - ... definido no contexto de uma declaração DOCTYPE

```
<!DOCTYPE elemento-raiz [declarações-de-elementos]>
```

- O DTD pode ser externo (“external reference”) ao documento XML
  - e estar escrito no ficheiro nome-ficheiro

```
<!DOCTYPE elemento-raiz SYSTEM "nome-ficheiro">
```



# Documento XML com especificação DTD (“inline”)

DTD

XML - conteúdo

```
<?xml version = "1.0" encoding="ISO-8859-1" standalone="yes"?>
<!DOCTYPE listaDeBares [
  <!ELEMENT listaDeBares (bar*)>
  <!ELEMENT bar (nome, cerveja+)>
  <!ELEMENT nome (#PCDATA)>
  <!ELEMENT cerveja (nome, preco)>
  <!ELEMENT preco (#PCDATA)>
]
>
```

Declaração XML

Declaração  
Tipo Documento

Declarações DTD

Elemento Documento  
ou elemento raiz

```
<listaDeBares>
  <bar>
    <nome>O Bar do Inglês</nome>
    <cerveja><nome>Bud</nome><preco>2.50</preco></cerveja>
    <cerveja><nome>Miller</nome><preco>3.00</preco> </cerveja>
  </bar>
  <bar>
    <nome>O Pinguim</nome>
    <cerveja><nome>Antarctica</nome><preco>2.00</preco></cerveja>
  </bar>
</listaDeBares>
```

# XML: visto na perspectiva da linguagem DTD

- Perspectiva do programador: os blocos de construção do XML são:
  - ❑ as marcas (“tags”)
  - ❑ e.g. `<bar><nome>O Pinguim</nome></bar>`
- Perspectiva do DTD: os blocos de construção do XML são:
  - ❑ elemento (e.g. `<nome>O Pinguim</nome>`)
    - estrutura base de um documento XML (e HTML)
  - ❑ atributo (e.g. `<bar nome="O Pinguim"/>`)
    - informação extra sobre um elemento
  - ❑ entidade (e.g. `&amp; ≡ &`)
    - variável usada para conter texto
  - ❑ PCDATA (“parsed character data”)
    - texto que irá ser analisado pelo *parser* (análise de marcas)
  - ❑ CDATA (Unparsed “character data”)
    - texto que não irá ser analisado pelo *parser* (marcas não são tratadas como tal)

DTD é *case sensitive*

# Declarações DTD: elementos

- Os elementos XML são especificados com uma declaração DTD

- ❑ `<!ELEMENT nome categoria>`, ou
- ❑ `<!ELEMENT nome (tipo-de-conteudo-do-elemento)>`

- Diferentes tipos de declarações de elemento

- ❑ **vazio**
  - `<!ELEMENT nome EMPTY>`
- ❑ **contém qualquer tipo de conteúdo (definido no DTD)**
  - `<!ELEMENT nome ANY>`
- ❑ **não contém elementos (apenas sequência de caracteres)**
  - `<!ELEMENT nome (#PCDATA)>`
- ❑ **contém elementos descendentes (nível imediatamente inferior)**
  - `<!ELEMENT nome (nome-elem-1, nome-elem-2?, nelem3*, (nelem4, nelem5)+, (nelem6 | nelem7) )>`
    - ❑ A sequencialidade tem de ser respeitada
    - ❑ Nome-elem2 é opcional, nelem3 pode ocorrer 0 ou mais vezes, (nelem4, nelem5) pode ocorrer 1 ou mais vezes, seguido de nele 6 ou nelem7
  - `<!ELEMENT note (#PCDATA|to|from|header|message) *>`
    - ❑ Note pode conter várias ocorrências dos itens referidos

PCDATA ≡ “parsed character data”

Caracteres que são examinados pelo *parser*, procurando elementos e outros símbolos

# Declarações DTD: atributos

- Os atributos XML são especificados com uma declaração DTD
  - `nome-elem`  $\equiv$  nome do elemento que tem o atributo `nome-atr`
  - `<!ATTLIST nome-elem (nome-atr tipo-atr valor-omissao) +>`

Tipo de atribto	Representa
CDATA	Conteúdo textual (não verificado)
NMTOKEN	Um Nmtoken (*)
NMTOKENS	Lista de NMTOKEN (**)
ENTITY	Referência para uma DTD entity
ENTITIES	Lista de entidades
ID	Um Xml Name de valor único
IDREF	Valor de um outro atributo ID
IDREFS	Lista de IDREFs
(e1   e2   ...)	um de vários Nmtokens (ei)
NOTATION	nome de uma anotação
xml:	valor XML predefinido

Que valor-omissao usar?	O que representa?
"valor"	valor omissão do atributo
#REQUIRED	atributo obrigatório
#IMPLIED	atributo opcional
#FIXED "valor"	a existir, deve ter o valor "valor"

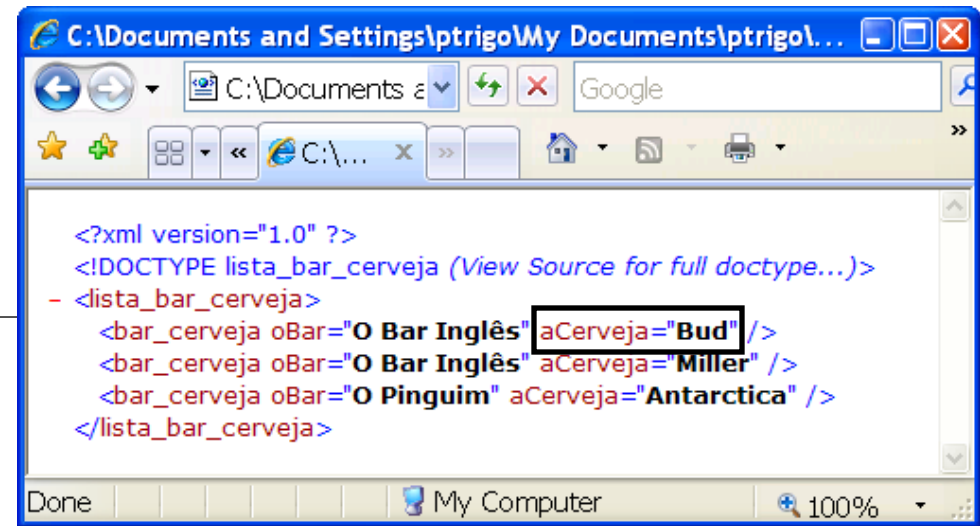
(\*)  
**XML Name**  
Name ::= NameStartChar (NameChar)\*  
NameStartChar ::= ":" | [A-Z] | "\_" |  
[a-z]  
NameChar ::= NameStartChar | "-" | "."  
| [0-9]  
  
**XML Nmtoken**  
Nmtoken ::= (NameChar)+

(\*\*) dois NMTOKEN separam-se por espaço.

# Declarações DTD: elementos e atributos (exemplo)

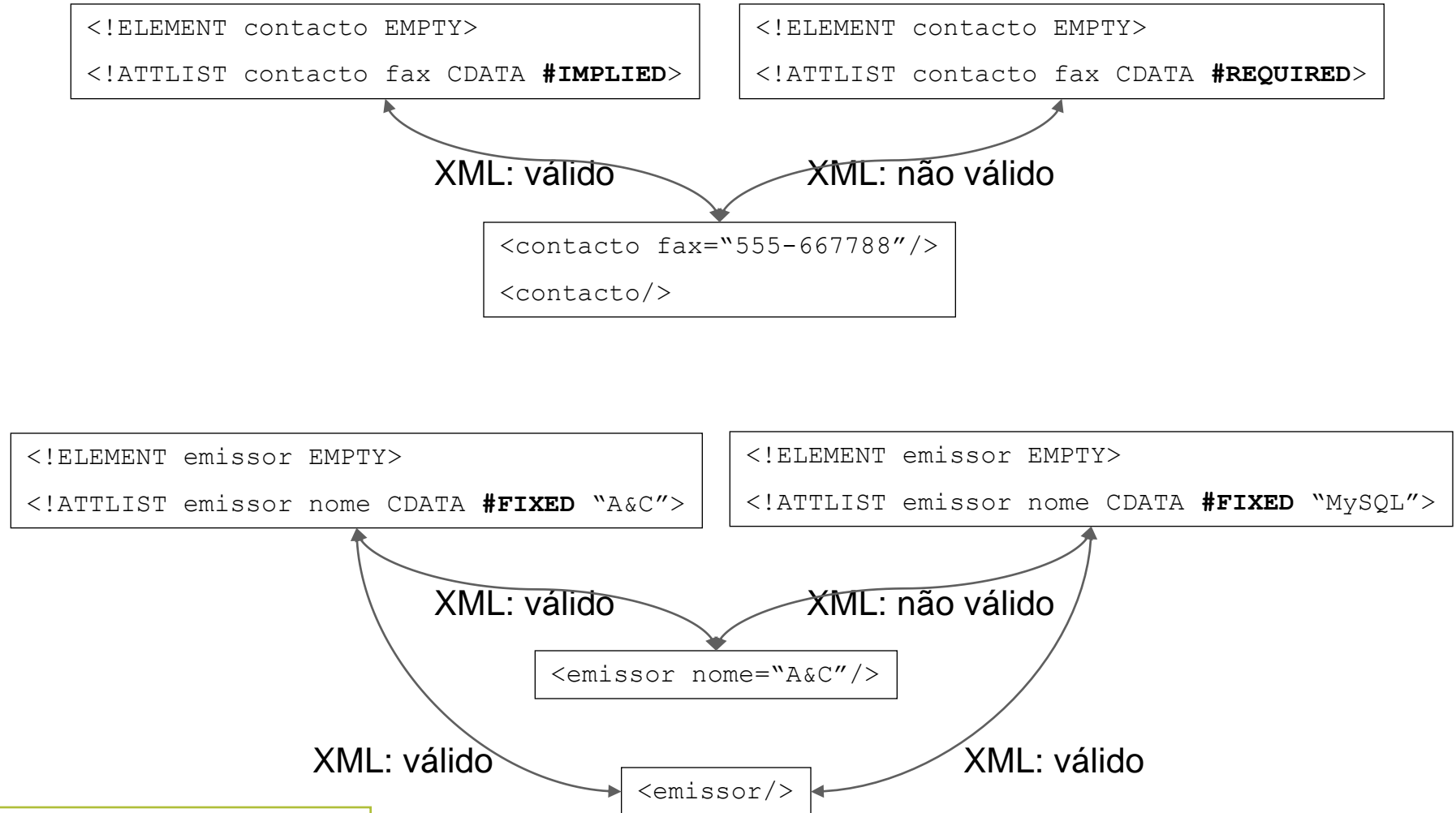
A lista `bar_cerveja` associa cada nome de bar (obrigatório) a um nome de cerveja que por omissão é “Bud”!

```
<?xml version="1.0"?>
<!DOCTYPE lista_bar_cerveja
[
  <!ELEMENT lista_bar_cerveja (bar_cerveja*)>
  <!ELEMENT bar_cerveja EMPTY>
  <!ATTLIST bar_cerveja oBar CDATA #REQUIRED>
  <!ATTLIST bar_cerveja aCerveja CDATA "Bud">
]>
<lista_bar_cerveja>
  <bar_cerveja oBar="O Bar Inglês"/>
  <bar_cerveja oBar="O Bar Inglês" aCerveja="Miller"/>
  <bar_cerveja oBar="O Pinguim" aCerveja="Antarctica"/>
</lista_bar_cerveja>
```



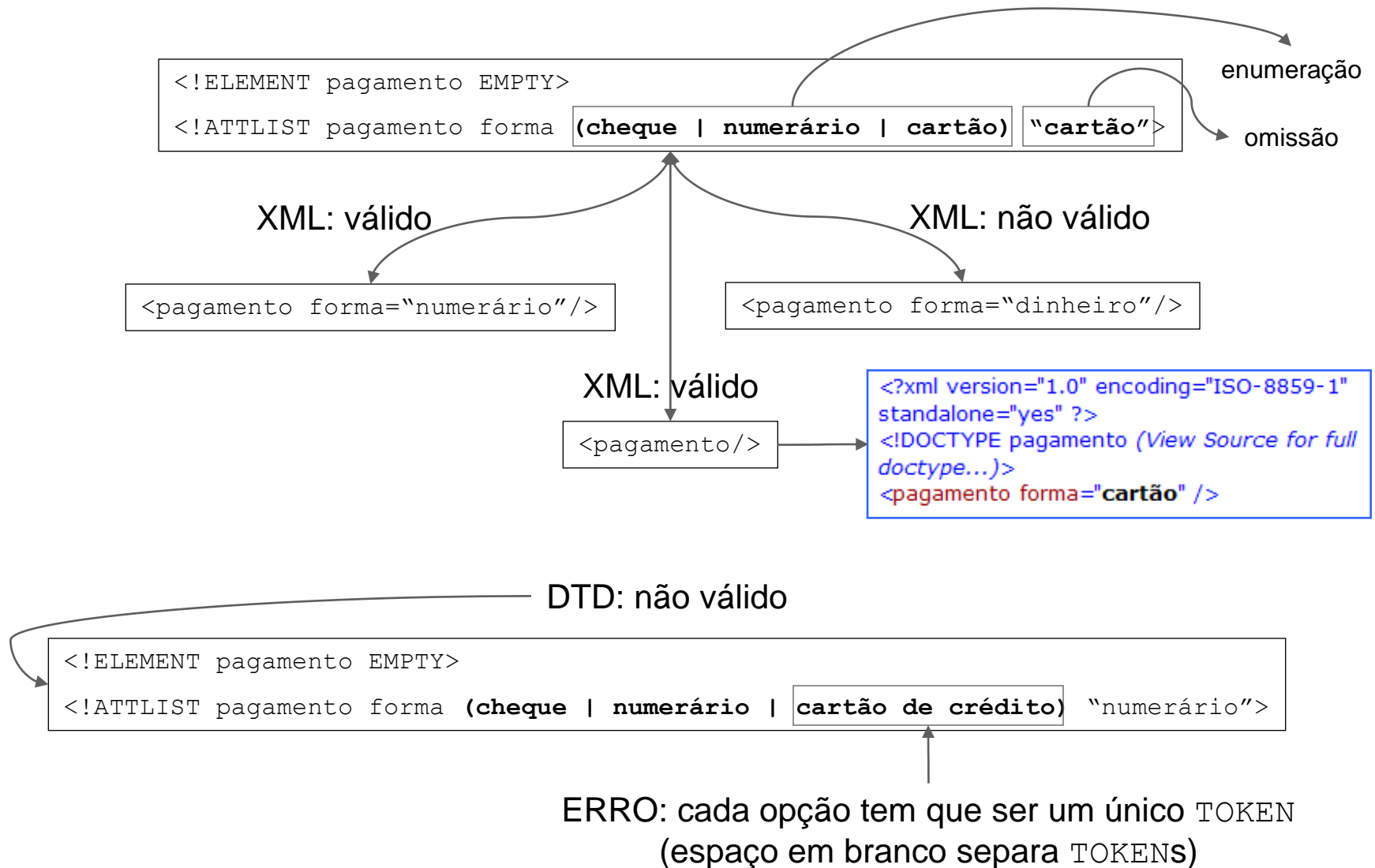
```
<!ATTLIST bar_cerveja
  oBar CDATA #REQUIRED
  aCerveja CDATA "Bud"
>
```

# DTD e XML: atributo IMPLIED, REQUIRED, FIXED



Estas verificações o IE 7 não as faz

# DTD e XML: atributo com enumeração de valores



---

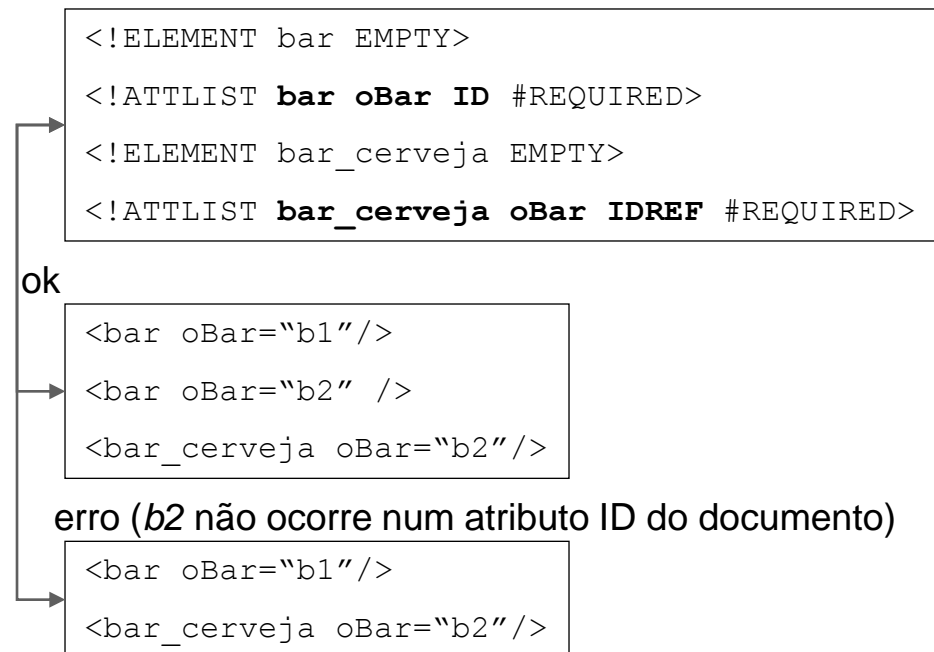
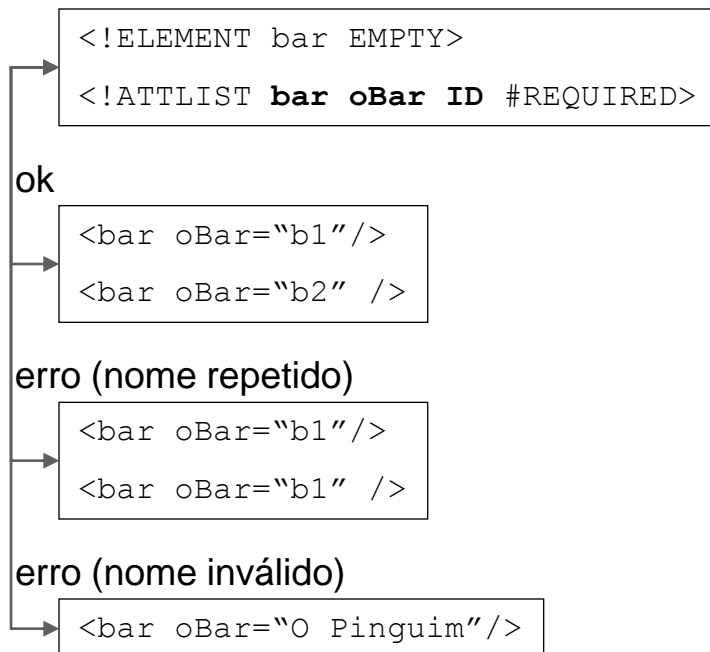
“ID” e “IDREF”

# RESTRIÇÕES DE INTEGRIDADE



# DTD: permite definir “restrições de integridade”?

- O DTD permite impor, para o contexto de um documento XML,
  - algumas “restrições de integridade” entre valores
- As restrições usam-se para obrigar
  - a unicidade de valores de um atributo (tipo ID), e
  - a referência (tipo IDREF) a valores que ocorram no documento



# Atributo de valor único (ID)

- Um atributo de valor único tem tipo ID
  - pode conter, como valor, um qualquer nome válido XML
    - ... tem que ser nome XML e não um “token” XML ( $\neq$  no 1º carácter)
    - e.g. nomes inválidos: .NET (inicia com .); O Pinguim (espaço)
  - e está garantido que esse nome é único no documento
- Cada elemento apenas pode ter um atributo do tipo ID
  - e.g. não é possível indicar unicidade de valores de um par de atributos
- ... recordar: um número não é um nome válido XML!
  - ... e.g. 12 é “token” mas não nome válido XML (inicia com dígito)
  - Um nome válido XML, apenas pode começar por letra ou underscore
- Cuidado ao usar números como valores de atributos de tipo ID
  - a solução usual é a de prefixar o valor por uma letra ou “underscore”
  - e.g. b12 ou \_12 (ambos são nomes válidos XML)

# Atributo referência (IDREF) para valor único (ID)

- Um atributo referência (para outro de tipo ID) tem tipo IDREF
  - apenas pode conter o valor de um atributo de tipo ID
- No entanto, o IDREF não indica qual o atributo ID que refere
  - no atributo IDREF é válido o valor de qualquer atributo de tipo ID!
  - ... não há garantia de integridade entre quem refere e quem é referido!

```
<!ELEMENT bar EMPTY>
<!ATTLIST bar oBar ID #REQUIRED>

<!ELEMENT cerveja EMPTY>
<!ATTLIST cerveja aCerveja ID #REQUIRED>

<!ELEMENT bar_cerveja EMPTY>
<!ATTLIST bar_cerveja oBar IDREF #REQUIRED>
<!ATTLIST bar_cerveja aCerveja IDREF #REQUIRED>

<!ELEMENT tudoJunto EMPTY>
<!ATTLIST tudoJunto bares IDREFS #REQUIRED>
<!ATTLIST tudoJunto cervejas IDREFS #REQUIRED>
```

```
<bar oBar="b1"/>
<bar oBar="b2"/>

<cerveja aCerveja="c1"/>
<bar_cerveja oBar="c1" aCerveja="b1"/>
<bar_cerveja oBar="b2" aCerveja="c1"/>

<tudoJunto bares="b1 b1" cervejas="c1 b2">
```

ok (no entanto, em bar\_cerveja, c1 e b1 estão trocados!), e refs

---

“Entity” e “Parameter Entity”

**ENTITY**

# Entidade (“entity”) DTD

- Uma entidade é uma variável usada para conter texto
  - ❑ Declaração: `<!ENTITY nome-entidade “valor-entidade”>`
  - ❑ pode ser declarada interna (dentro) ou externa (fora) ao documento XML
- Referência a entidade
  - ❑ resulta na substituição da referência pelo valor da referida entidade
  - ❑ Sintaxe: `&nome-entidade`
- Entidade interna – declarada dentro do ficheiro
  - ❑ Na parte DTD: `<!ENTITY NOME “Haruki Murakami”>`
  - ❑ Na parte XML: `<autor>&NOME;</autor>`
- Entidade externa – será declarada noutro ficheiro, no ficheiro onde for utilizada terá de se utilizar:
  - ❑ Declaração: `<!ENTITY nome-entidade SYSTEM “URI”>`
  - ❑ e.g. `<!ENTITY NOME SYSTEM “nomeAutor.xml”>` (refere ficheiro)

NOME será substituído pelo valor da entidade referenciada

Ficheiro onde a entidade está definida a entidade

# Exemplo de utilização de Entities

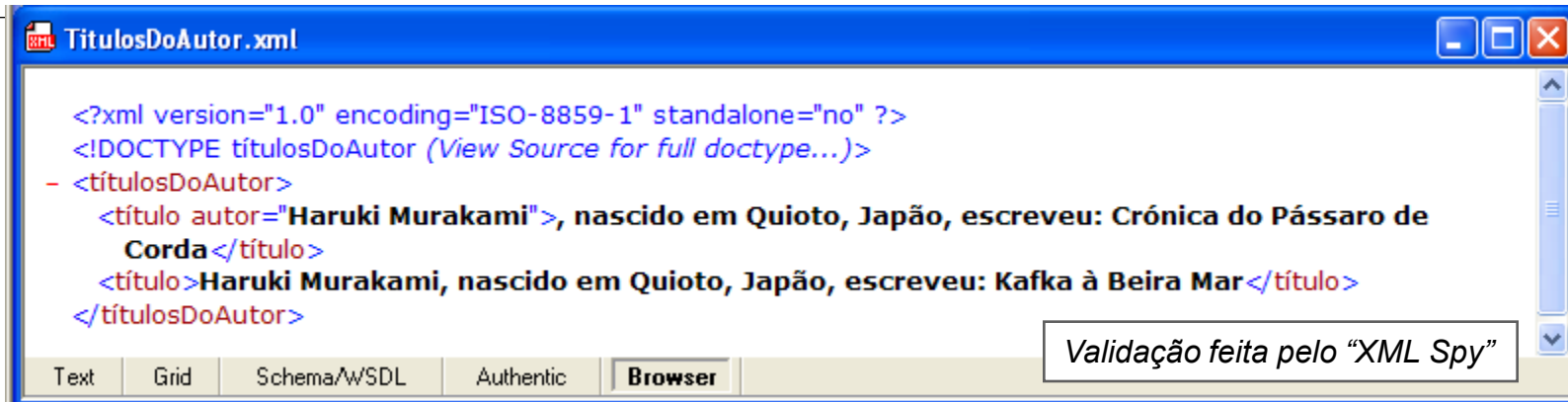
```
<!DOCTYPE títulosDoAutor
[ <!ENTITY AUTOR "Haruki Murakami">
  <!ENTITY LOCAL_NASC "Quioto, Japão,">
  <!ENTITY T1 "nascido em">
  <!ENTITY T2 "escreveu">
  <!ENTITY TEXTO "&AUTOR;, &T1; &LOCAL_NASC; &T2;:">
  <!ELEMENT títulosDoAutor (título*)>
  <!ELEMENT título (#PCDATA)>
  <!ATTLIST título autor CDATA #IMPLIED> ]>

<títulosDoAutor>

  <título autor="&AUTOR;", &T1; &LOCAL_NASC; &T2;: Crónica do Pássaro de Corda</título>

  <título>&TEXTO; Kafka à Beira Mar</título>

</títulosDoAutor>
```



# DTD: subconjunto interno (“internal subset”)

- O subconjunto interno (“internal subset”) do DTD consiste
  - ❑ nas definições que estão no mesmo ficheiro que o documento XML
  - ❑ ... boa abordagem para iniciar a construção do DTD (facilita alterações)

```
<?xml version="1.0"?>

<!DOCTYPE lista_bar_cerveja
[
  <!ELEMENT lista_bar_cerveja (bar_cerveja*)>
  <!ELEMENT bar_cerveja EMPTY>
  <!ATTLIST bar_cerveja oBar CDATA #REQUIRED>
  <!ATTLIST bar_cerveja aCerveja CDATA "Bud">
]>

<lista_bar_cerveja>
  <bar_cerveja oBar="O Bar Inglês"/>
  <bar_cerveja oBar="O Bar Inglês" aCerveja="Miller"/>
  <bar_cerveja oBar="O Pinguim" aCerveja="Antarctica"/>
</lista_bar_cerveja>
```

**subconjunto interno**

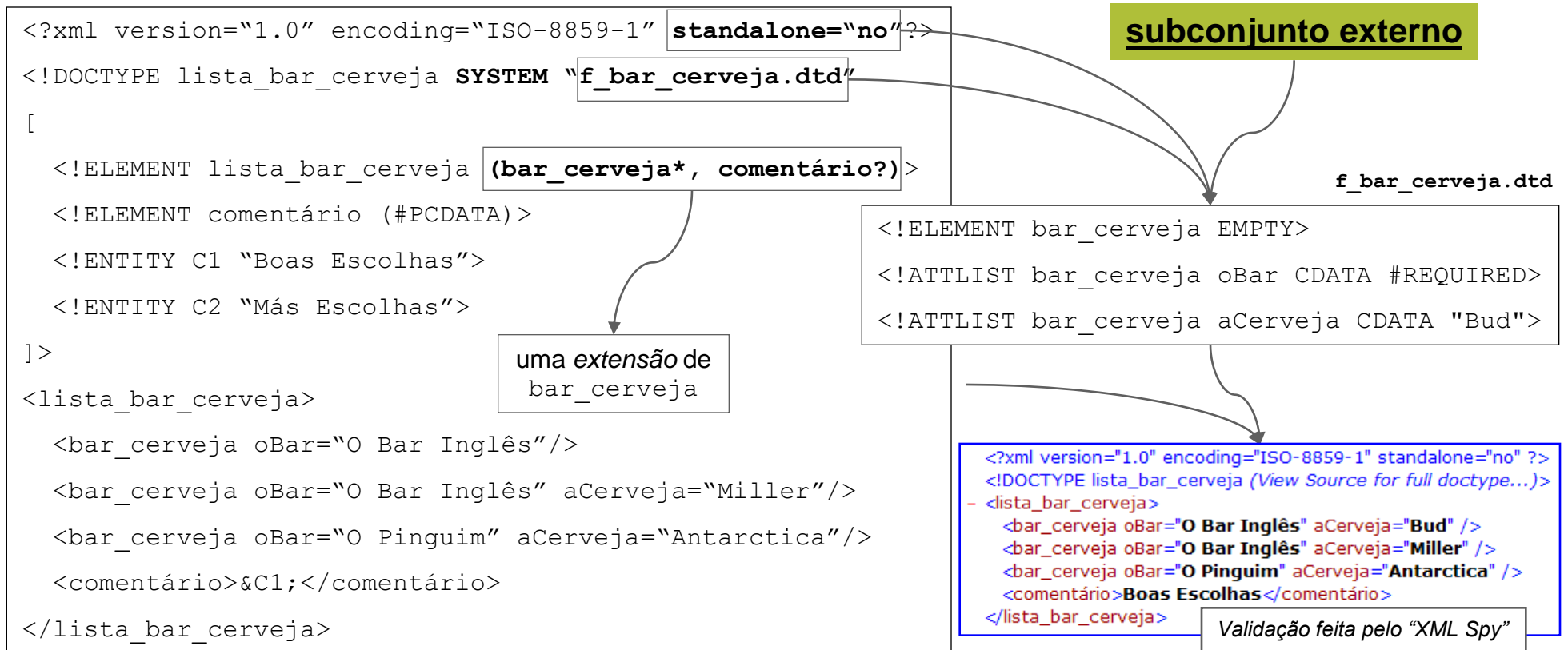
... Mas para partilhar o DTD  
implica  
isolar o DTD noutro ficheiro  
(i.e. criar subconjunto externo)

... em geral o DTD pode ter:

- um subconjunto interno
- um subconjunto externo

# DTD: subconjunto externo (“external subset”)

- O subconjunto externo (“external subset”) do DTD consiste
  - nas definições que estão num ficheiro diferente do documento XML
  - ... boa abordagem para partilhar e estender DTDs





# DTD: subconjuntos interno e externo

- O subconjunto interno (“internal subset”) do DTD está contido
  - entre os parêntesis rectos: `<!DOCTYPE r [ subconjunto interno ]>`
- O subconjunto externo (“external subset”) do DTD estão no
  - ficheiro referido em SYSTEM
    - Ficheiro `f.dtd`, em: `<!DOCTYPE r SYSTEM "f.dtd">`
  - na declaração XML deve-se ter: `standalone="no"`
- Quando existe subconjunto interno e externo tem-se
  - `<?xml version="1.0" standalone="no"?>`
  - `<!DOCTYPE r SYSTEM "filename.dtd" [ subconjunto interno ]>`
- Os subconjuntos interno e externo têm que ser compatíveis
  - nenhum pode redefinir os elementos e atributos do outro
  - ... mas as declarações de entidades (ENTITY) podem ser redefinidas!
    - Prevalecem as definições de entidades do conjunto interno

# Entidade parâmetro

- Uma entidade parâmetro (“parameter entity”) consiste
  - numa entidade geral
  - que depois se utiliza para definir elementos ou atributos
  - Só é válida no contexto DTD\* (não válida no conteúdo XML)
- Uma entidade parâmetro declara-se como a ENTITY com sinal %
  - `<!ENTITY % nome-entidade "valor-entidade">`
  - e.g. `<!ENTITY % comum "autor+, título, ano?">`
  - ... e depois podem definir-se elementos usando esta entidade!
- Exemplo: “uma definição de livro e de artigo”

```
<!ELEMENT livro (autor+, título, ano?, editora?, ISBN?)>  
<!ELEMENT artigo (autor+, título, ano?, local?, tipoPublicação?)>
```

```
<!ELEMENT livro (%comum;, editora?, isbn?)>  
<!ELEMENT artigo (%comum;, local?, tipoPublicação?)>
```

\* E utilizadas em ficheiro DTD

# Características da entidade parâmetro

- Diferença entre entidade (geral) e entidade parâmetro
  - entidade (geral): referida com & ; entidade parâmetro: referida com %
  - entidade parâmetro: só pode ser usada no DTD (subconjunto externo)
- Exemplo: “na bibliografia constam diversos livros e artigos”
  - têm pelo menos 1 autor, 1 título e eventualmente o ano de publicação
  - o livro tem ainda a editora e o ISBN
  - o artigo tem o local em que foi publicado e o tipo de publicação

```
<!ENTITY % comum "autor+, título, ano?">

<!ELEMENT bibliografia (livro | artigo)*>

<!ELEMENT livro (%comum;, editora?, ISBN?)>

<!ELEMENT artigo (%comum;, local?, tipoPublicação?)>
```

f\_comum.dtd

# Entidade parâmetro (exemplo)

```
<!ENTITY % comum "autor+, título, ano?">
<!ELEMENT bibliografia (livro | artigo)*>
<!ELEMENT livro (%comum;, editora?, isbn?)>
<!ELEMENT artigo (%comum;, local?, tipoPublicação?)>
<!ENTITY % atrnome 'nome'>
<!ATTLIST bibliografia %atrnome; CDATA #IMPLIED>
```

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>

<!DOCTYPE bibliografia SYSTEM "f_comum.dtd"
[
  <!ELEMENT autor (#PCDATA)> <!ELEMENT título (#PCDATA)>
  <!ELEMENT ano (#PCDATA)>   <!ELEMENT editora (#PCDATA)>
  <!ELEMENT isbn (#PCDATA)>   <!ELEMENT local (#PCDATA)>
  <!ELEMENT tipoPublicação (#PCDATA)>
]>

<bibliografia nome="dd">
  <livro><autor>a1</autor><título>t1</título></livro>
  <artigo><autor>a2</autor><título>t2</título></artigo>
</bibliografia>
```



Validação feita pelo "XML Spy"

# Redefinição de entidades

- Os analisadores (“parsers”) percorrem o DTD pela seguinte ordem,
  - primeiro: subconjunto interno, e
  - segundo: subconjunto externo
- Quando uma mesma entidade tem mais que uma definição
  - é considerada a primeira definição encontrada

... logo, as entidades DTD do subconjunto interno podem redefinir as entidades DTD do subconjunto externo.

Atenção (recordar):

Pode-se redefinir-se entidades.

Não se pode redefinir elementos, nem atributos.

- Redefinir entidades parâmetro permite, por exemplo
  - Alterar, incluir ou excluir definições

# Alterações às definições de elementos e atributos

- **Pode-se indicar que uma definição pode ser ignorada ou incluída:**
  - ❑ **[IGNORE[ definição ]]**, para ignorar uma definição
  - ❑ **[INCLUDE[ definição ]]**, para incluir uma definição
- ... “na lista de bares e cervejas não existia o atributo `preço`”
  - ❑ ... mas nas novas listas já consta esse atributo (e é obrigatório)!
  - ❑ objectivo: contemplar o atributo com o mínimo de alterações ao DTD
  - ❑ ... como configurar a indicação (ignora ou inclui) a dar ao analisador?

o DTD original (a alterar o mínimo!)

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!ELEMENT lista_bar_cerveja (bar_cerveja*)>
<!ELEMENT bar_cerveja EMPTY>

<!ATTLIST bar_cerveja oBar CDATA #REQUIRED>
<!ATTLIST bar_cerveja aCerveja CDATA "Bud">
```

Para que o analisador ignore a declaração

```
<![ IGNORE [
<!ATTLIST bar_cerveja preço CDATA #REQUIRED>
]]>
```

para que o analisador inclua a declaração

```
<![ INCLUDE [
<!ATTLIST bar_cerveja preço CDATA #REQUIRED>
]]>
```

# Exemplo: incluir ou ignorar um atributo na definição DTD

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT lista_bar_cerveja (bar_cerveja*)>
<!ELEMENT bar_cerveja EMPTY>
<!ATTLIST bar_cerveja oBar CDATA #REQUIRED>
<!ATTLIST bar_cerveja aCerveja CDATA "Bud">
<![%SOBRE_O_PREÇO; [ <!ATTLIST bar_cerveja preço CDATA #REQUIRED> ]]>
```

f\_lista\_bares.dtd

alteração  
do DTD

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE lista_bar_cerveja SYSTEM "f_lista_bares.dtd"
[ <!ENTITY % SOBRE_O_PREÇO "INCLUDE"> ]>

<lista_bar_cerveja>
  <bar_cerveja oBar="O Bar Inglês" preço="3.00"/>
  <bar_cerveja oBar="O Bar Inglês" aCerveja="Miller" preço="2.50"/>
  <bar_cerveja oBar="O Pinguim" aCerveja="Antarctica" preço="2.00"/>
</lista_bar_cerveja>
```

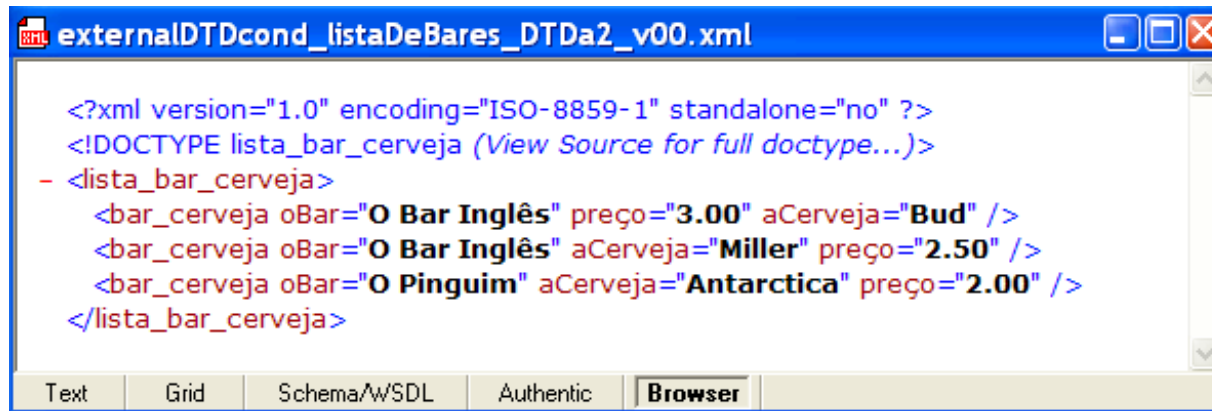
```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE lista_bar_cerveja SYSTEM "f_lista_bares.dtd"
[ <!ENTITY % SOBRE_O_PREÇO "IGNORE"> ]>
<lista_bar_cerveja>
  <bar_cerveja oBar="O Bar Inglês"/>
  <bar_cerveja oBar="O Bar Inglês" aCerveja="Miller"/>
  <bar_cerveja oBar="O Pinguim" aCerveja="Antarctica"/>
</lista_bar_cerveja>
```

... a nova lista com o preço

... a lista antiga sem o preço

# Exemplo: resultados do analisador (“parser”) XML

... a nova lista com o preço



The screenshot shows a window titled 'externalDTDcond\_listaDeBares\_DTDa2\_v00.xml'. The XML content is as follows:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<!DOCTYPE lista_bar_cerveja (View Source for full doctype...)>
- <lista_bar_cerveja>
  <bar_cerveja oBar="O Bar Inglês" preço="3.00" aCerveja="Bud" />
  <bar_cerveja oBar="O Bar Inglês" aCerveja="Miller" preço="2.50" />
  <bar_cerveja oBar="O Pinguim" aCerveja="Antarctica" preço="2.00" />
</lista_bar_cerveja>
```

The 'preço' attribute is present in all three beer entries. The bottom toolbar shows 'Text', 'Grid', 'Schema/WSDL', 'Authentic', and 'Browser' (selected).

... a lista antiga sem o preço



The screenshot shows a window titled 'externalDTDcond\_listaDeBares\_DTDa1\_v00.xml'. The XML content is as follows:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<!DOCTYPE lista_bar_cerveja (View Source for full doctype...)>
- <lista_bar_cerveja>
  <bar_cerveja oBar="O Bar Inglês" aCerveja="Bud" />
  <bar_cerveja oBar="O Bar Inglês" aCerveja="Miller" />
  <bar_cerveja oBar="O Pinguim" aCerveja="Antarctica" />
</lista_bar_cerveja>
```

The 'preço' attribute is missing from all three beer entries. The bottom toolbar shows 'Text', 'Grid', 'Schema/WSDL', 'Authentic', and 'Browser' (selected).

Validação feita pelo “XML Spy”



# O que são: entidade não analisada e anotação?

- Como incluir informação que não adere às convenções do XML?
  - e.g. uma fotografia em formato GIF, um ficheiro de som MIDI, etc
- Definir 2 entidade externas não analisadas (ENTITY e NOTATION)
  - uma com a informação pretendida e atribuição de nome ao tipo de dados
    - `<!ENTITY umaFoto SYSTEM "foto.gif" NDATA gif>`
  - outra indicando o formato do tipo de dado considerado
    - `<!NOTATION gif SYSTEM "image/gif">`
    - ... semelhante à "instrução de processamento"

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foto[
  <!ENTITY umaFoto SYSTEM "foto.gif" NDATA gif>
  <!NOTATION gif SYSTEM "image/gif">
  <!ELEMENT foto EMPTY>
  <!ATTLIST foto imagem ENTITY #REQUIRED>]>
<foto imagem="umaFoto"/>
```

Não há norma para indicar o formato;  
e.g. "image/gif" indica um formato  
binário MIME.

Nota: não usar & pois é  
entidade não analisada.

Para se ver a imagem pode-se utilizar uma transformada XSLT.

# DTD: CONCLUSÕES

# O que o DTD fornece e o que não fornece...

- O DTD permite especificar
  - ❑ a estrutura (hierárquica) dos elementos
  - ❑ restrições sobre o número de ocorrências de cada elemento
  - ❑ atributos permitidos em cada elemento
  - ❑ tipos gerais de atributo e valor de omissão
- O DTD não permite especificar
  - ❑ detalhe sobre o tipo de atributo (e.g. `dateTime` em vez de `CDATA!`)
  - ❑ detalhe sobre restrições de ocorrência (e.g. nº exacto de ocorrências)
  - ❑ detalhe sobre restrições entre diferentes ocorrências...

# Qual o próximo passo?

- Para simples páginas “Web” o DTD pode ser suficiente, mas
  - para aplicações baseadas em dados (e.g. “web service”)
  - é preciso um controlo mais rigoroso sobre o conteúdo dos documentos
- O XSD (“XML Schema Definition”) fornece esse rigor!
  - é baseado na sintaxe do XML
  - suporta diferentes tipos de dados
  - suporta “espaços de nomes” para integração de diferentes convenções
- ... o XSD é um forte candidato a substituto do DTD
  - em aplicações “Web”!